

Dokumen Teknis

Aplikasi Mobile Klinik Del

Tugas Proyek Akhir Semester

Mata Kuliah: Pengembangan Aplikasi Terdistribusi

Dipersiapkan oleh:

NIM 11322038	Samuel Albi Pulo S
NIM 11322039	Kenan Tomfie Bukit
NIM 11320051	Sarah Meilani Butar Butar
NIM 11320061	Johanna Romauli Siagian

Untuk:

**Institut Teknologi Del
2024**



**PROYEK PENGEMBANGAN APLIKASI TERDISTRIBUSI
INSTITUT TEKNOLOGI DEL 2022**

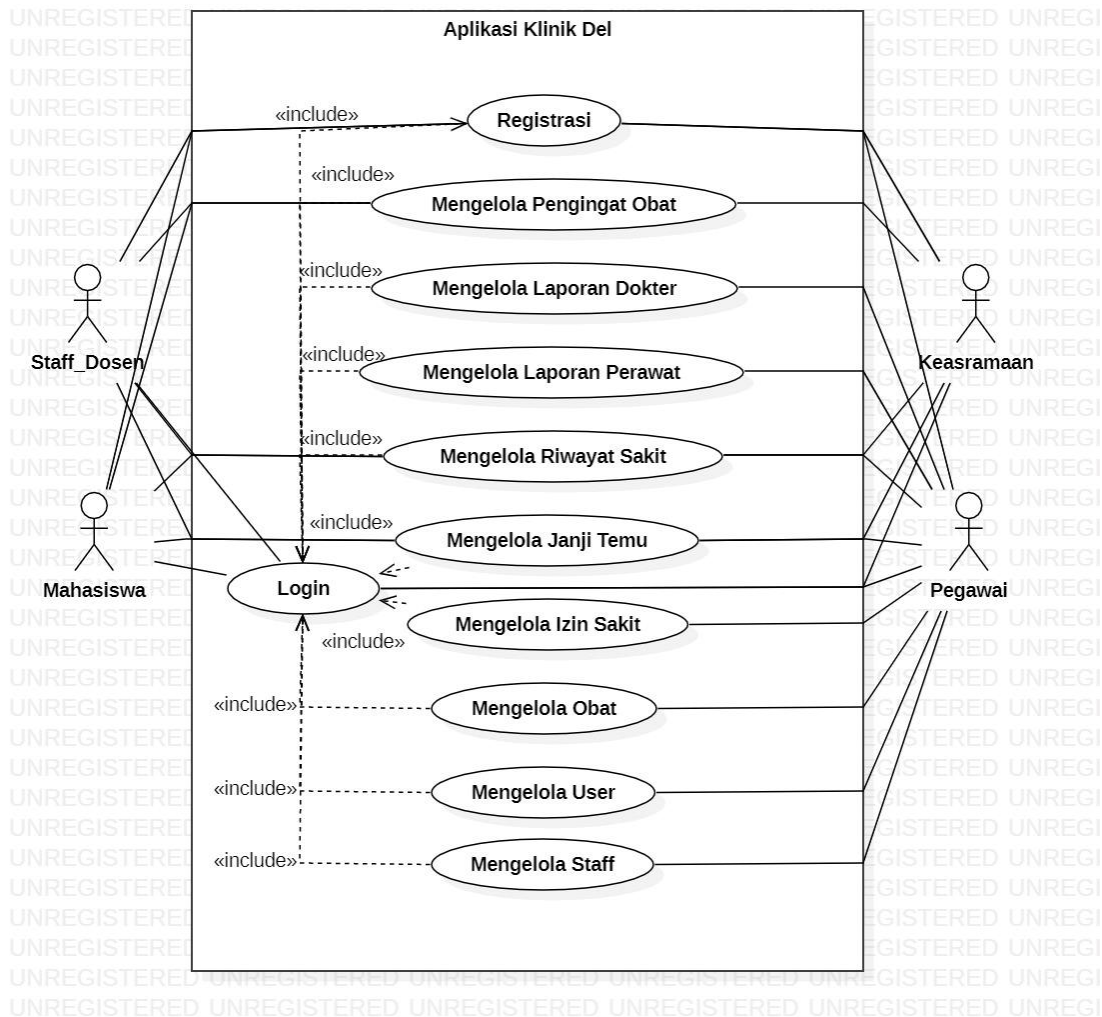
DAFTAR ISI

1	Deskripsi Umum Aplikasi	1
2	Deskripsi Fungsi	4
3	Arsitektur Monolitik	8
4	Arsitektur Microservice	14
	4.1 Tampilan UI.....	14
	4.2 Pull & Push Model.....	14

1 Deskripsi Umum Aplikasi

Pembangunan aplikasi ini ditujukan untuk untuk klinik Institut Teknologi Del. Aplikasi ini dapat digunakan oleh pegawai klinik dalam layanan klinik dan mahasiswa untuk membuat antrian secara online. Pada aplikasi mobile, mahasiswa dapat membuat janji temu untuk datang ke klinik sesuai dengan waktu yang sudah dibuat, sehingga mahasiswa tidak perlu antri lagi ketika sampai di klinik. Website klinik digunakan pegawai perawat dan dokter untuk mengelola data mahasiswa atau staff yang berobat ke klinik. Aplikasi klinik dibangun menggunakan 3 bahasa pemrograman. Bagian back-end dan front-end dibangun menggunakan bahasa yang berbeda. Untuk bagian front-end menggunakan dart dan javascript, sedangkan back-end menggunakan go-lang dengan framework fiber.

2 Deskripsi Fungsi



Berikut merupakan deskripsi fungsi yang ada dalam klinik Del :

- **Fungsi Mengelola Janji Temu**
Fungsi ini digunakan oleh mahasiswa dan staf Del untuk membuat janji temu dengan dokter. Pada fungsi ini, mereka dapat mengatur jadwal pertemuan dengan dokter sesuai kebutuhan.
- **Fungsi Mengelola Report Dokter**
Fungsi ini digunakan oleh dokter untuk mengelola laporan mengenai mahasiswa dan staf yang melakukan konsultasi. Dokter dapat mencatat hasil pemeriksaan, diagnosis,

serta rekomendasi pengobatan dalam sistem.

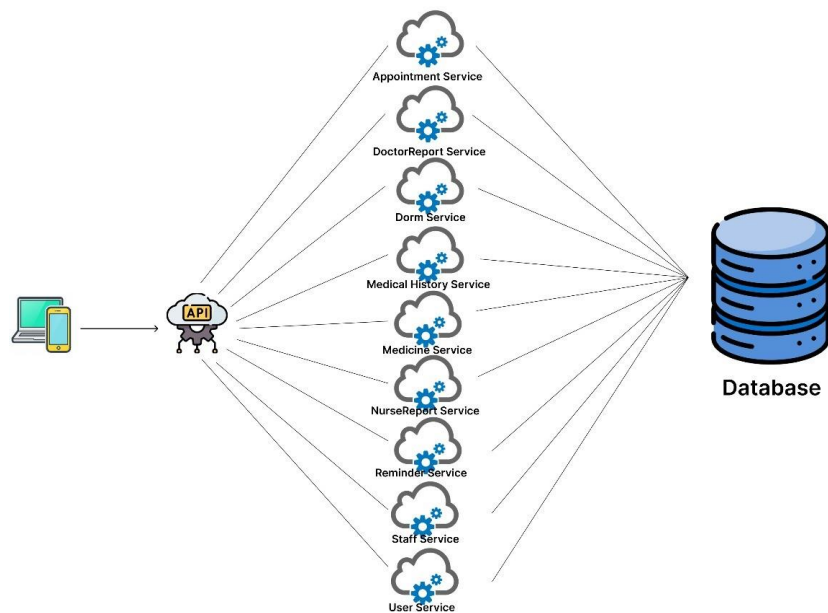
- **Fungsi Mengelola Asrama**
Fungsi tersebut bertujuan untuk mengidentifikasi penempatan setiap mahasiswa ke dalam asrama yang sesuai.
- **Fungsi Mengelola Riwayat Sakit**
Fungsi ini berperan dalam menampilkan catatan mengenai riwayat penyakit yang dialami oleh seorang pasien setelah menjalani sesi konsultasi medis. Informasi ini membantu dalam perencanaan dan pemantauan perawatan kesehatan pasien.
- **Fungsi Mengelola Obat**
Fungsi ini digunakan untuk mengelola persediaan obat, termasuk pengadaan, penyimpanan, distribusi, dan pencatatan penggunaan obat dalam aplikasi.
- **Fungsi Mengelola Report Suster**
Fungsi ini digunakan oleh suster untuk mengelola laporan mengenai mahasiswa dan staf yang melakukan konsultasi. Suster dapat mencatat informasi vital pasien, memberikan perawatan dasar, serta mengupdate status kesehatan pasien
- **Fungsi Mengelola Pengingat Obat**
Fungsi ini berperan dalam menyediakan sistem atau layanan yang membantu individu dalam mengingat jadwal konsumsi obat mereka dengan tepat waktu. Pengingat ini dapat disesuaikan dengan jadwal konsumsi obat masing-masing pasien.
- **Fungsi Mengelola Data Staff**
Fungsi ini digunakan untuk mengelola data staff yang terdaftar dalam aplikasi, termasuk informasi pribadi, riwayat kerja, dan hak akses ke fitur-fitur tertentu dalam sistem.
- **Fungsi Mengelola Data User**
Fungsi ini digunakan untuk mengelola data pengguna yang terdaftar dalam aplikasi, seperti informasi kontak, preferensi pengingat, dan histori aktivitas pengguna dalam sistem.

3 Arsitektur Monolitik

Arsitektur monolitik ini merupakan cara yang paling mudah dan cepat untuk membuat sistem, karena semua komponen berada dalam satu tempat. Dalam pembangunan aplikasi klinik Del menggunakan arsitektur monolitik, dapat dilihat pada gambar dibawah semua fitur berada dalam satu aplikasi tunggal yang dibuat menggunakan satu bahasa dan dijalankan melalui satu server. Database yang digunakan tunggal menyimpan semua tabel tiap fungsi.

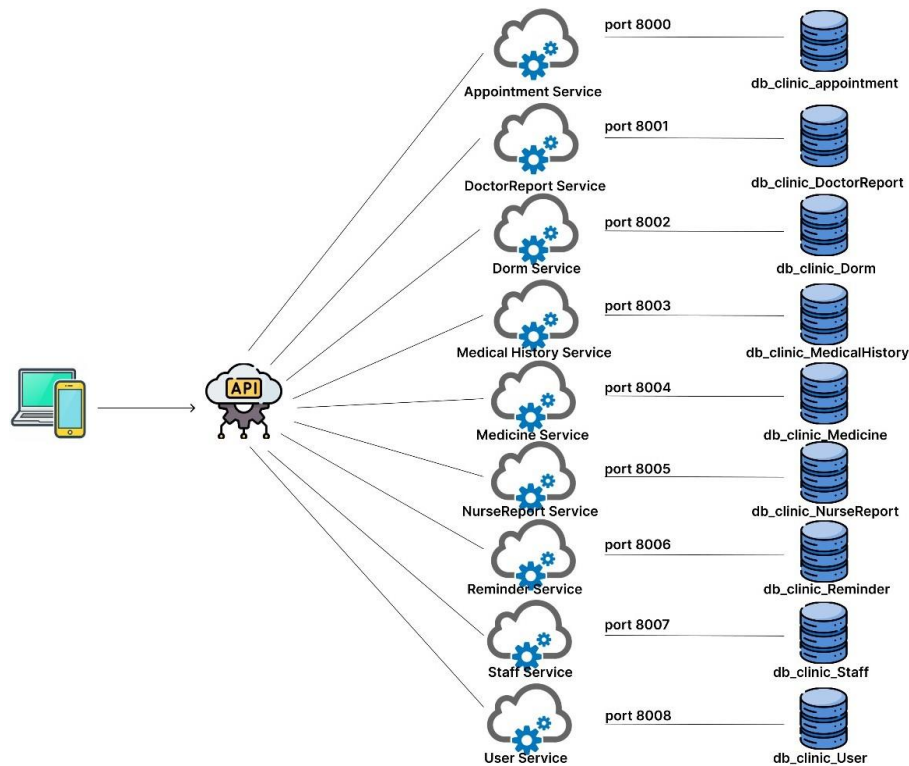
Kelebihan arsitektur monolitik adalah lebih mudah dikembangkan, di-debug dan di-maintain karena semua komponen tersedia dalam satu aplikasi.

Kekurangan dari arsitektur monolitik adalah lebih sulit untuk di-scaling secara parsial. Jika salah satu bagian dari aplikasi mengalami masalah, maka seluruh aplikasi akan bermasalah juga. Sehingga, aplikasi dengan arsitektur monolitik menjadi kompleks dan sulit untuk dikembangkan. Arsitektur monolitik tidak fleksibel terhadap perubahan bisnis karena harus mengubah semua aplikasi yang sudah di implementasi.



4 Arsitektur Microservices

Arsitektur microservices adalah model arsitektur yang memisahkan aplikasi menjadi beberapa komponen independen yang dapat dikembangkan dan diubah tanpa mempengaruhi komponen lainnya. Berbeda dengan arsitektur monolitik, setiap komponen dalam arsitektur microservices dapat diimplementasikan berdiri sendiri, dikembangkan, di-deploy, dan diubah secara independen.

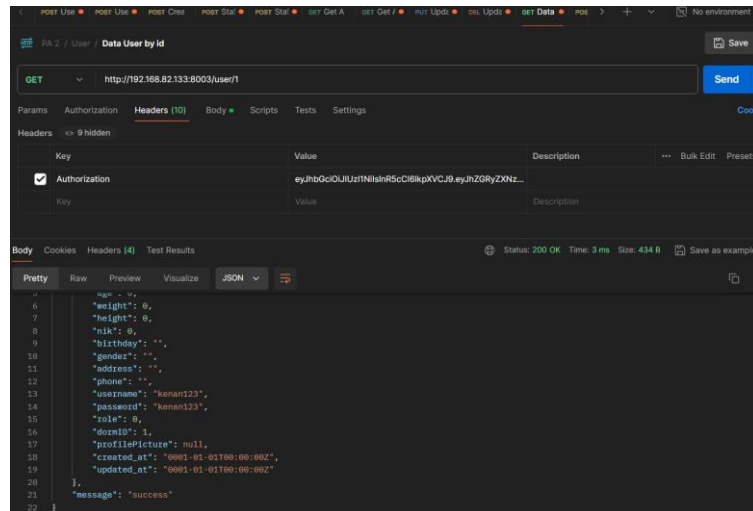


Pada gambar diatas merupakan arsitektur microservice yang digunakan dalam pengembangan aplikasi klinik Del.

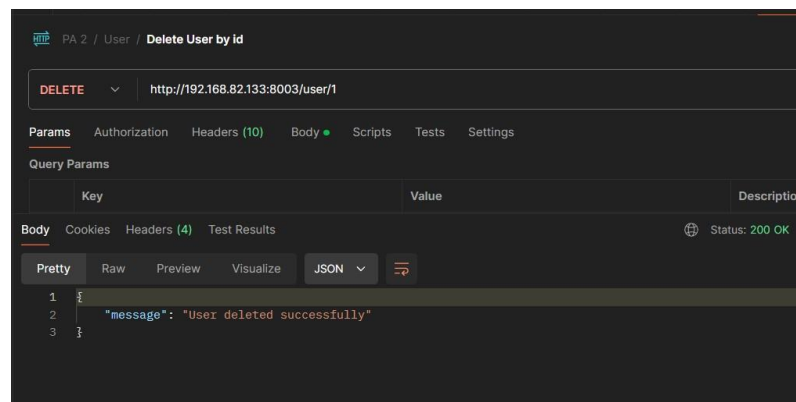
Setiap service dalam aplikasi klinik Del dipisah, sehingga memiliki database-nya masing-masing. Pada aplikasi ini terdapat 9 service dengan 9 database yaitu : *Appointment*, *DoctorReport*, *Dorm*, *MedicalHistory*, *Medicine*, *NurseReport*, *Reminder*, *Staff*, *User*.

Service akan berjalan secara paralel dan tidak mempengaruhi service lain jika terdapat masalah. REST API digunakan sebagai penghubung antara *web ui* dengan *service* yang tersedia sehingga memungkinkan *service* dapat dikonsumsi oleh *web ui*. Hal inilah yang menyebabkan *service* dapat diakses oleh beberapa *platform* tanpa harus terbatas pada satu bahasa pemrograman saja.

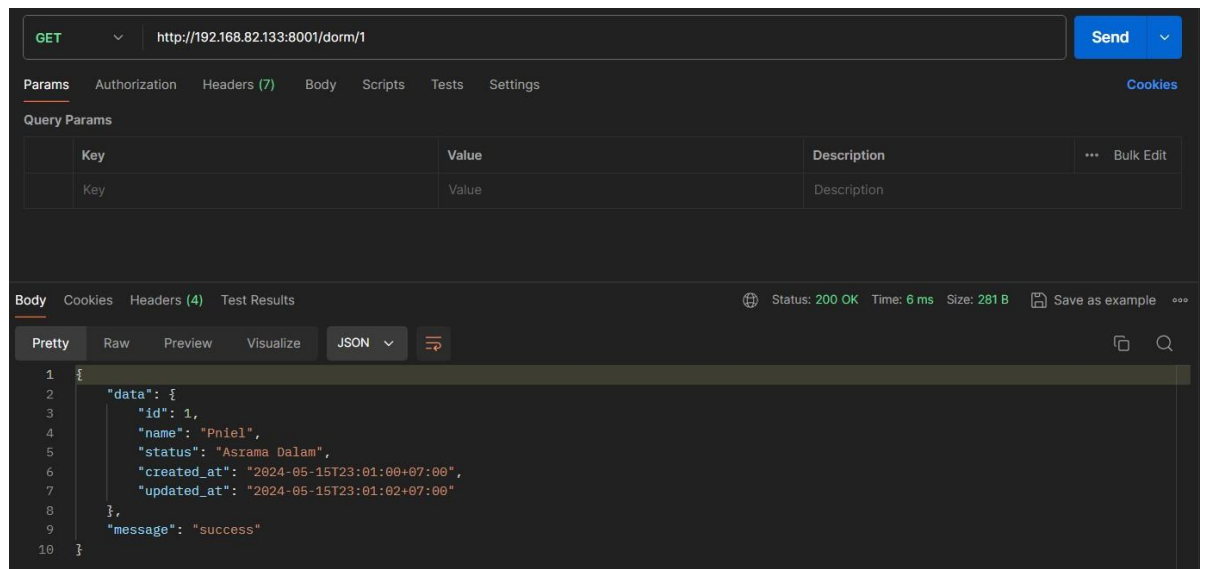
Kelebihan arsitektur microservice adalah lebih mudah untuk di-scaling secara parsial. Jika salah satu bagian dari aplikasi mengalami masalah, hanya bagian tersebut yang terpengaruh, sehingga mengurangi dampak pada seluruh sistem. Arsitektur ini lebih mudah untuk dikembangkan dan diperbarui, karena setiap komponen dapat dikembangkan dan diperbarui secara independen.



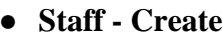
• User - delete



• Dorm



• Staff - Create



```
POST http://192.168.82.133:8007/staff

{
  "username": "hasan1234",
  "password": "hasan1234"
}
```

Status: 200 OK Time: 819 ms Size: 476 B Save as example

Pretty Raw Preview Visualize JSON

```
{
  "data": {
    "id": 3,
    "name": "",
    "age": 0,
    "weight": 0,
    "height": 0,
    "NIP": 0,
    "birthday": "",
    "gender": "",
    "address": "",
    "phone": "",
    "username": "hasan1234",
    "password": "$2a$14$yksh.EqbUMGSLurhPonaYuIiY9adByJwbPk3PjyG5w2L0d1jo/MLu",
    "role": 0,
    "profilePicture": null,
    "created_at": "2024-05-15T23:24:21.608+07:00",
    "updated_at": "2024-05-15T23:24:21.608+07:00"
  },
  "message": "success"
}
```

• Appointment - Create

```
POST http://192.168.82.133:8000/appointment

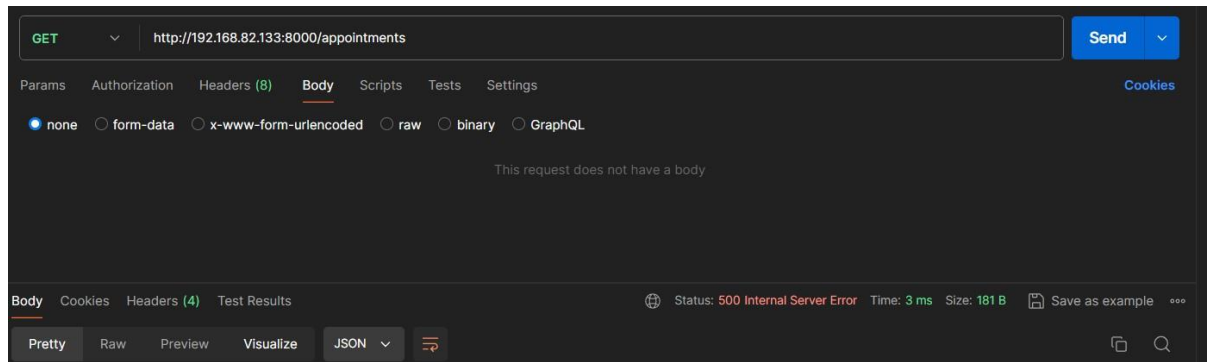
{
  "date": "12 mei 2022",
  "time": "10:00",
  "complaint": "mencret",
  "requested_id": 1
}
```

Status: 200 OK Time: 10 ms Size: 388 B Save as example

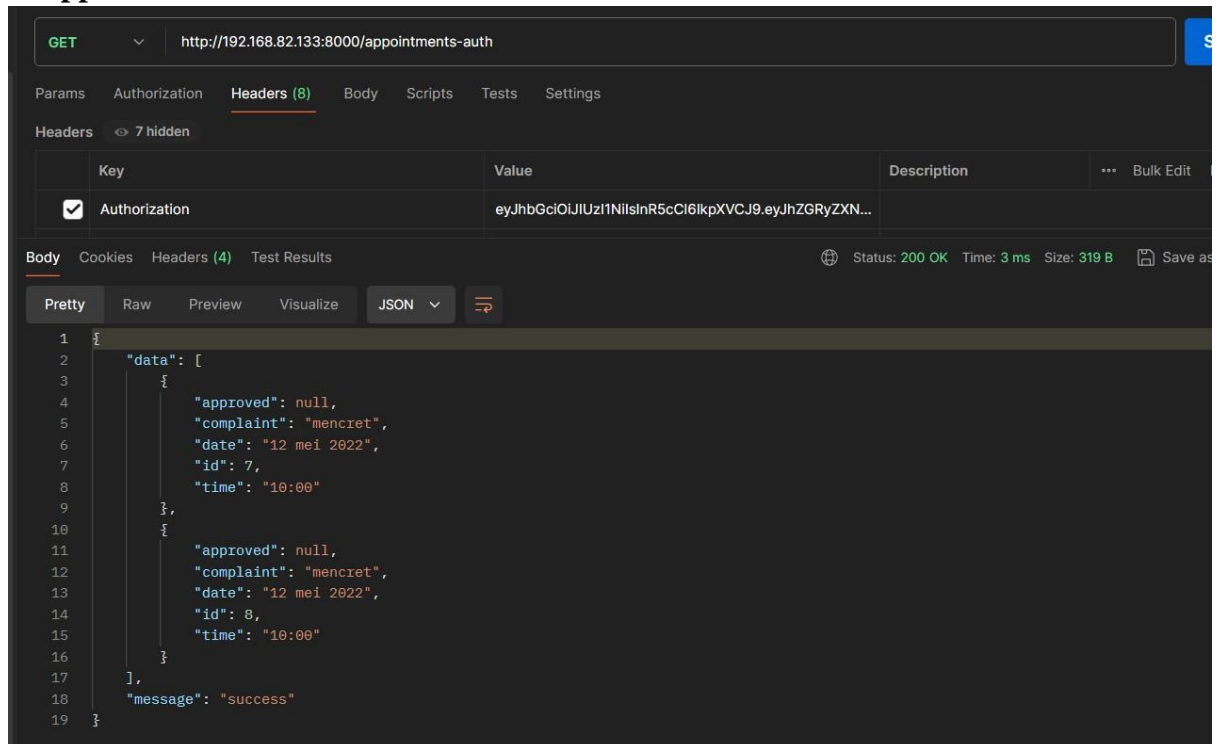
Pretty Raw Preview Visualize JSON

```
{
  "appointment": {
    "id": 3,
    "date": "12 mei 2022",
    "time": "10:00",
    "complaint": "mencret",
    "approved_id": null,
    "requested_id": 1,
    "created_at": "2024-05-15T23:35:37.921+07:00",
    "updated_at": "2024-05-15T23:35:37.921+07:00"
  },
  "message": "created data successfully",
  "status": "success"
}
```

• Appointment - Get All



- Appointment - Auth



- Appointment - Update

PUT ▼ http://192.168.82.133:8000/appointment/:id

Params ● Authorization Headers (10) **Body** ● Scripts Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1 {
2   "date": "12 mei 2022",
3   "time": "10:00",
4   "complaint": "mencret parah"
5 }
```

Body Cookies Headers (4) Test Results ⊕ Status

Pretty Raw Preview Visualize **JSON** ▼ ≡

```
1 {
2   "data": {
3     "id": 8,
4     "date": "12 mei 2022",
5     "time": "10:00",
6     "complaint": "mencret parah",
7     "approved_id": null,
8     "requested_id": 2,
9     "created_at": "2024-05-15T23:39:23.451+07:00",
10    "updated_at": "2024-05-15T23:42:48.628+07:00"
11  },
12  "message": "Success"
13 }
```

- **Appointment - Delete**

DELETE ▼ http://192.168.82.133:8000/appointment/:id Send ▼

Params ● Authorization Headers (10) **Body** ● Scripts Tests Settings Cookies

Key	Value	Description
-----	-------	-------------

Path Variables

Key	Value	Description	...	Bulk Edit
id	5	Description		

Body Cookies Headers (4) Test Results ⊕ Status: 200 OK Time: 7 ms Size: 168 B Save as example ...

Pretty Raw Preview Visualize **JSON** ▼ ≡ 📄 🔍

```
1 {
2   "message": "Appointment deleted successfully"
3 }
```

- **Appointment - Approved by Clinic**

PUT

http://192.168.82.133:8000/appointment/:id

Params

Authorization

Headers (10)

Body

Scripts

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
2  {
3    "date": "12 mei 2022",
4    "time": "10:00",
5    "complaint": "mencret",
6    "approved_id": 3,
7    "requested_id": 2
8  }
```

Body

Cookies

Headers (4)

Test Results

Status: 200 OK

Time: 92 ms

Size: 343 B

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "data": {
3      "id": 8,
4      "date": "12 mei 2022",
5      "time": "10:00",
6      "complaint": "mencret",
7      "approved_id": null,
8      "requested_id": 2,
9      "created_at": "2024-05-15T23:39:23.451+07:00",
10     "updated_at": "2024-05-15T23:47:47.96+07:00"
11   },
12   "message": "Success"
13 }
```

